5. Candy

Program Name: Candy.java Input File: candy.dat

Halloween is just around the corner, and you are only now getting around to going shopping for candy!

Not a complete slacker, you've been thorough in your research and have calculated how many neighborhood children will come to your door. Fortunately (or unfortunately), they travel in a single pack of candy-craving goblins and ghouls.

Each child should get an equal amount of candy, otherwise they'll begin to squabble, throw tantrums, and wreak general havoc. According to your deductions, there are *n* possible group sizes of children that can come to your door.

You want to make sure that no matter how many children come to your door, you will have enough to split the candy evenly among them all. Since you waited so late to go candy shopping, all the good stuff was sold out at every grocery store in the city (how unlucky!), so you want to make sure that you have NO candy left after distributing it on Halloween, otherwise you'd have to eat subpar candy, and nobody wants that.

What is the smallest amount of candy you have to buy in order to split up the candy evenly to any possible group of children without having any left to spare?

Input
The first line of input consists of a single integer, *t*, indicating the number of test cases that follow.

For each test case, the first line will consist of a single integer, *n*, indicating how many potential group sizes there may be on Halloween. On the next line, there will be *n* integers, which represent all of the potential group sizes.

Output
For each test case, print on its own line the minimum number of candy pieces you should buy.

Constraints
```
1 <= t <= 8
1 <= n <= 10

All group sizes will be under 100. The answer will always fit within a 32-bit
signed integer.
```

Example Input File
```
2
2
3 4
3
5 10 12
```

Example Output to Screen
```
12
60
```